CpSc 2120 — Goddard — Spring'18

Assignment 1

Due 8pm Sunday 11 February

Encryption by Letter Substitution. Caesar Julius used to send mesages by first reversing them and then applying a rotation. The ***rotation*** operation for a string replaces each letter by the next letter in the alphabet, with Z being replaced by A. Example: When Zebra is rotated, one gets Afcsb. Note that punctuation and spaces are ignored in the rotation, and the case is preserved. The goal is to produce a program that can decipher a sentence that has been rotated an unknown number of times. This entails finding the "most likely" rotation.

1) Initial class MySentence

You should produce a MySentence class (stored in .cpp and .h of the same name). This stores a sentence. It should initally have at least the following:

- a constructor that takes a string
- a reverse function that reverses the contents
- a static rotationOf(char) function that returns the rotated char. It should preserve the case.
- a rotate function that rotates the stored sentence once
- a rotate(int) function that rotates the stored sentence a specified number of times
- a test for equality with another MySentence
- overloaded stream insertion operator for output

2) Class Corpus

In order to do the decoding, we need to know the relative proportions of characters in English. Create a class Corpus (stored in files of the same name) that determines the proportions. It should have the following:

- a no-argument constructor which sets the proportions according to the Scrabble set: that is, the proportions are 0.09, 0.02, 0.02, 0.04, 0.12, 0.02, 0.03, 0.02, 0.09, 0.01, 0.01, 0.04, 0.02, 0.06, 0.08, 0.02, 0.01, 0.06, 0.04, 0.06, 0.04, 0.02, 0.02, 0.01, 0.02, 0.01 (even though these only sum to 0.98)
- a constructor that takes the name of a text file and uses that to determine the proportions
- a function proportion(char) that takes a lower-case char and returns the proportion associated with that.

The constructor should echo the 26 proportions to the standard output. It is recommended that initially you code up only the no-arguments constructor.

3) Doing the decoding

Implement a primitive decoding method. For this, you need to add two functions to your MySentence class.

- double scoreWith(Corpus&): this sums the proportions for each letter in the current sentence. For example, if the proportions are the default ones, and scoreWith is invoked with Cab-DA, then it should return 0.26 (the calculation is 0.02+0.09+0.02+0.04+0.09). Punctuation, spaces, digits etc. are ignored.
- void decode(Corpus&): this steps through all 26 possible rotations. For each rotated sentence, it calculates the scoreWith and keeps track of the highest value (and which rotation gave you that value). After considering all rotations, it sets itself to the rotated string that gave the highest score.

4) Driver file

Create a driver file Runner.cpp that reads the file name, the rotated sentence, invokes the decode function, and then prints out the result.

Here is sample run: Make sure that your output is similar.

```
Enter file name for corpus:  juliusCaesar.txt
Frequencies are 0.078, 0.018, 0.027, 0.038, 0.129, 0.020, 0.015, 0.056, 0.065,
                0.001, 0.008, 0.043, 0.028, 0.061, 0.080, 0.015, 0.000, 0.063,
                0.071, 0.089, 0.043, 0.005, 0.022, 0.002, 0.024, 0.000,
Enter sentence terminated by <ENTER> !gnwt utgikV
Decoded sentence is: Tigers rule!
```

5) Submission

Submit your source code using handin. You are to work independently, but can ask questions from the lecturer and lab instructor(s). Late submissions will be significantly penalized.